# CSC1135 Secure Programming Lab 04

Darragh O'Brien B.Sc. Ph.D. FHEA
School of Computing
Dublin City University

2025-26

## Overview

In this lab we will:

- Do some string handling in C
- Look at static linking
- Look at dynamic linking

## A. String copying

Take a copy of copy.c and do the following:

- Identify the problem with the program
- Fix the program by modifying the `if` statement
- Fix the program by using `strncpy` rather than `strcpy`
- Modify the program to use `malloc` rather than a fixed-size buffer
- Modify the program to use `strdup` rather than a fixed-size buffer
- Modify the program to use `strndup` rather than a fixed-size buffer
- Ensure your code reports any problems encountered
- Here is my attempt

## B. String concatenation

Take a copy of the C source code in conc.c and do the following:

- Identify the problem with the program
- Fix the program by adding an `if` statement
- Fix the program using `strncpy` and `strncat`
- Fix the program using an `if` statement and `sprintf`
- Fix the program using `snprintf`
- Modify the program so it prompts for `firstname` and `surname`, uses `fgets` to read them and then securely concatenates them
- Ensure your code reports any problems encountered
- Here is my attempt

## C. Static linking

Using `objdump` to help you, identify the relocations that occur when linking foo.c and bar.c to form an executable `foobar`.

## D. Dynamic linking

Take copies of hook.c and hooked.c and follow the steps below:

```
$ gcc -shared -o hook.so hook.c -fPIC -ldl
$ gcc -o hooked hooked.c
$ LD_PRELOAD=`pwd`/hook.so ./hooked
```

Through running and studying the code can you work out what is going on?

Use this technique to hook calls to `strncpy`. Have the hook null-terminate the destination even where truncation occurs in copying from the source to the destination. Write a program to test your solution works.